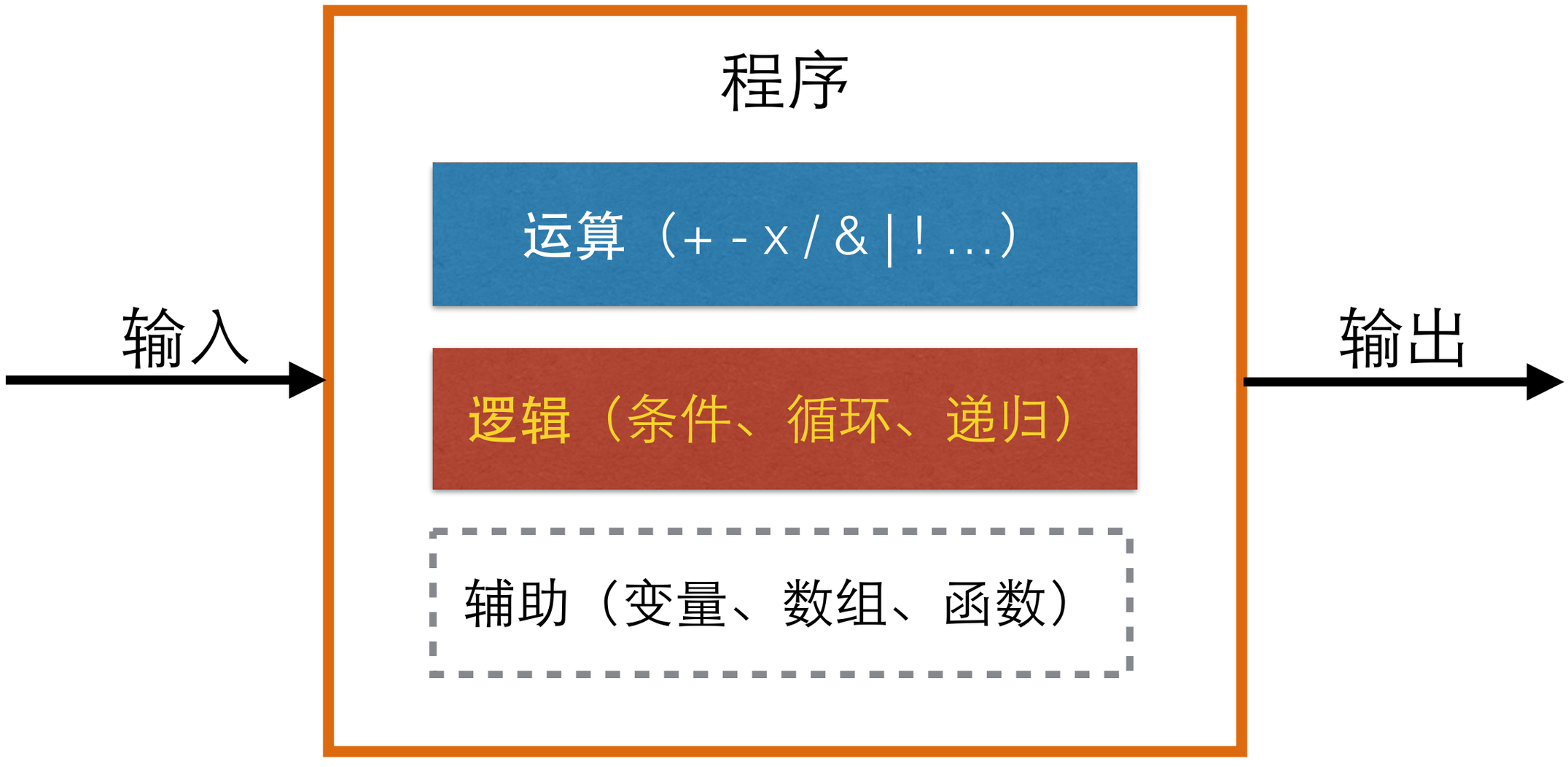
PHP基本语法
—条件、循环、函数

PHPCHINA! HAPPY PHPING
PHPCHINA.COM

杨亮

程序的基本结构



小测验

用你熟悉的程序找出1~1000中的所有质数

我们直接看代码好了

if

else

elseif

```php
<?php
if( $totalqty == 0 )
    echo 'You did not order anything on the previous page!<br />';

if ($totalqty == 0) {
    echo '<p style="color:red">';
    echo 'You did not order anything on the previous page!';
    echo '</p>';
}

if ($totalqty == 0) {
    echo "You did not order anything on the previous page!<br />";
} else {
    echo $tireqty." tires<br />";
    echo $oilqty." bottles of oil<br />";
    echo $sparkqty." spark plugs<br />";
}

if ($tireqty < 10) {
    $discount = 0;
} elseif (($tireqty >= 10) && ($tireqty <= 49)) {
    $discount = 5;
} elseif (($tireqty >= 50) && ($tireqty <= 99)) {
    $discount = 10;
} elseif ($tireqty >= 100) {
    $discount = 15;
}
```

switch

```php
if ($find == "a") {
    echo "<p>Regular customer.</p>";
} elseif ($find == "b") {
    echo "<p>Customer referred by TV advert.</p>";
} elseif ($find == "c") {
    echo "<p>Customer referred by phone directory.</p>";
} elseif ($find == "d") {
    echo "<p>Customer referred by word of mouth.</p>";
} else {
    echo "<p>We do not know how this customer found us.</p>";
}

switch($find) {
    case "a" :
        echo "<p>Regular customer.</p>";
        break;
    case "b" :
        echo "<p>Customer referred by TV advert.</p>";
        break;
    case "c" :
        echo "<p>Customer referred by phone directory.</p>";
        break;
    case "d" :
        echo "<p>Customer referred by word of mouth.</p>";
        break;
    default :
        echo "<p>We do not know how this customer found us.</p>";
        break;
}
?>
```

```
<html>
<body>
<table border="0" cellpadding="3">
    <tr>
        <td bgcolor="#CCCCCC" align="center">Distance</td>
        <td bgcolor="#CCCCCC" align="center">Cost</td>
    </tr>
    <tr>
        <td align="right">50</td>
        <td align="right">5</td>
    </tr>
    <tr>
        <td align="right">100</td>
        <td align="right">10</td>
    </tr>
    <tr>
        <td align="right">150</td>
        <td align="right">15</td>
    </tr>
    <tr>
        <td align="right">200</td>
        <td align="right">20</td>
    </tr>
    <tr>
        <td align="right">250</td>
        <td align="right">25</td>
    </tr>
</table>
</body>
</html>
```

Distance	Cost
50	5
100	10
150	15
200	20
250	25

```html
<html>
<body>
<table border="0" cellpadding="3">
    <tr>
        <td bgcolor="#CCCCCC" align="center">Distance</td>
        <td bgcolor="#CCCCCC" align="center">Cost</td>
    </tr>
```

```php
<?php
$distance = 50;
while ($distance <= 250) {
    echo "<tr>
            <td align=\"right\">".$distance."</td>
            <td align=\"right\">".($distance / 10)."</td> </tr>";
    $distance += 50;
}
?>
```

```html
</table>
</body>
</html>
```

Distance	Cost
50	5
100	10
150	15
200	20
250	25

while

```php
for ($distance = 50; $distance <= 250; $distance += 50) {
    echo "<tr>
            <td align=\"right\">".$distance."</td>
            <td align=\"right\">".($distance / 10)."</td> </tr>\n";
}
```

for

数组与循环

```
2 $prices = array('Tires'=>100, 'Oil'=>10, 'Spark
3
4 foreach ($prices as $key => $value) {
5     echo $key." — ".$value."<br />";
6 }
7
8 while ($element = each($prices)) {
9     echo $element['key'];
10    echo " — ";
11    echo $element['value'];
12    echo "<br />";
13 }
14
15 reset($prices);
16 while (list($product, $price) = each($prices)) {
17    echo "$product - $price<br />";
18 }
```

软件工程中的代码重用

- 因为我们都懒（做过的事情不想再做第二次）
- 降低开发成本
- 增加代码的可靠性
- 增强程序的一致性

PHP中的代码重用

- 将其他文件中的html或者php代码引入到本文件
- require()与include();
- require_once()与include_once();

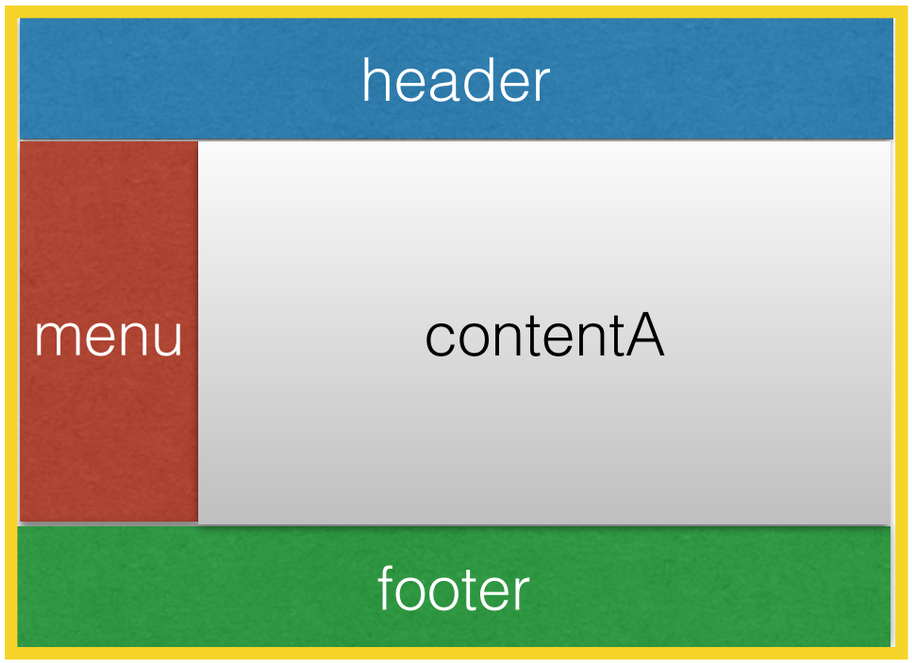
- 可以引入其他的函数库，或者代码片段

```php
<?php
//reusable.php
echo 'Here is a very simple PHP statement.<br />';
?>
```

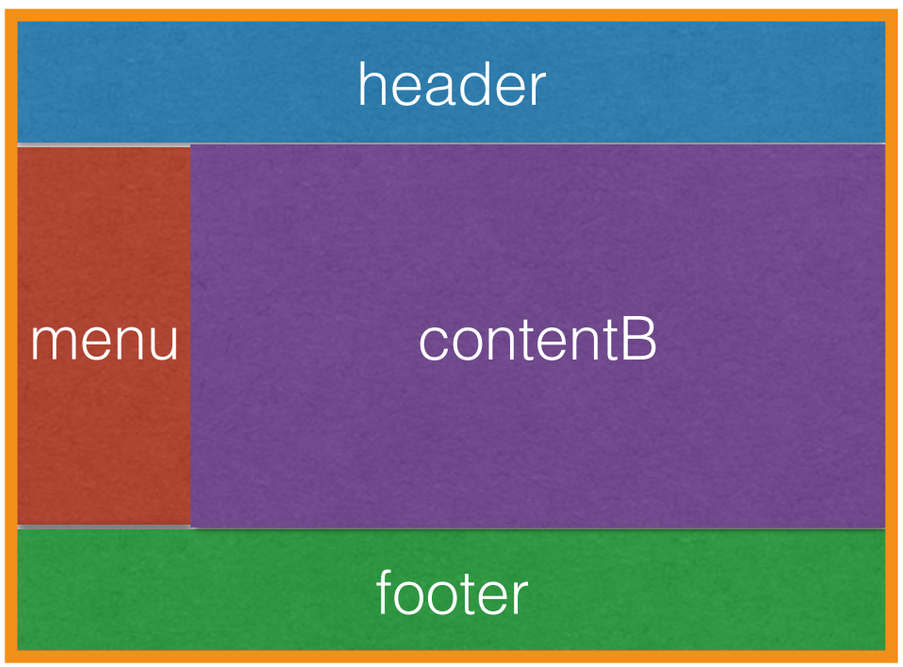
```php
<?php
//use.php
echo 'This is the main file.<br />';
require( 'reusable.php' );
echo 'The script will end now.<br /
?>
```

This is the main file.
Here is a very simple PHP statement.
The script will end now.

PHP实现网页模板




```
require('header.php');
require('menu.php');
contentA
require('footer.php');
```




```
require('header.php');
require('menu.php');
contentB
require('footer.php');
```

header.php

footer.php

menu
.php

找出1~1000中的所有质数

```php
<?php
for ($i=2; $i<1000; $i++) {
    for ($j=2; $j<sqrt($i); $j++) {
        if (doubleval($i)%doubleval($j) == 0 )
{
            break;
        }
    }
    if ($j > sqrt($i)) {
        echo $i.'<br>';
    }
}
?>
```

判断$i是否为质数

```php
<?php
for ($i=2; $i<1000; $i++)
    if (is_zhishu($i)) {
        echo $i.'<br>';
    }
}
?>
```

DRY: Don’t Repeat Yourself

~~Ctrl-C & Ctrl-V~~

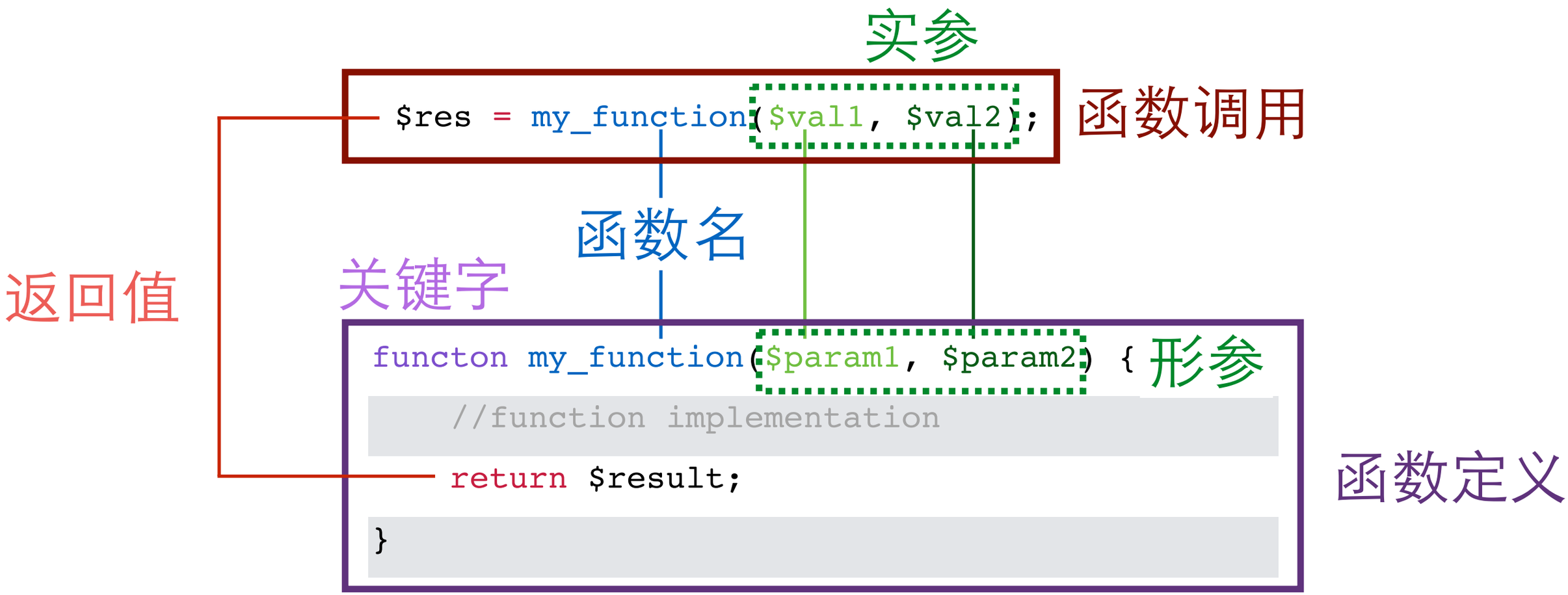
良好的代码风格

代码的可读性

为什么自己写有函数

- 代码的可读性
- 代码的可重用性
- 实现功能的模块化
- 实现递归调用
- 使变量名不至于太长（作用域）

PHP中的函数


实参
$res = my_function($val1, $val2);
函数调用
函数名
返回值
关键字
functon my_function($param1, $param2) {
形参
//function implementation
return $result;
}
函数定义

```php
for ($i=2; $i<1000; $i++) {
    for ($j=2; $j<sqrt($i); $j++) {
        if (doubleval($i)%doubleval($j) == 0 )
{
            break;
        }
    }
    if ($j > sqrt($i)) {
        echo $i.'<br>';
    }
}
```

```php
<?php
for ($i=2; $i<1000; $i++)
    if (is_zhishu($i)) {
        echo $i.'<br>';
    }
}
?>
```

```php
function is_zhishu($num) {
    for ($j=2; $j<sqrt($num); $j++) {
        if (doubleval($num)%doubleval($j) == 0 )
{
            break;
        }
    }
    if ($j > sqrt($num)) {
        return 1;
    }else{
        return 0;
    }
}
```

缺省参数值

在函数定义时指定缺省参数值

```
functon my_function($param1=default1, $param2=default2) {
    //function implementation
    return $result;
}
```

```
$res = my_function($val1, $val2);     //$param1=$val1, $param2=$val2
$res = my_function($val1); //$param1=$val1, $param2=default2
$res = my_function(); //$param1=default1, $param2=default2
```

缺省参数按照重要性排序，越可以自动确定的越放到后面

可变参数函数

- 所有输入值求和
- 所有输入值的最大公约数（最小公倍数）
- fun_num_args()与fun_get_args();

```php
<?php
function var_args() {
    echo "Number of parameters:";
    echo func_num_args();
    echo "<br />";
    $args = func_get_args();
    foreach ($args as $arg) {
        echo $arg."<br />";
    }
}
?>
```

变量作用域

- 局部变量（函数中定义），从定义处到函数结束
- 全局变量（非函数中定义），从定义出到文件尾，但不包括其中的函数
- 超全局变量（下节课会讲到），所以地方
- require和include都不影响变量作用域，只是起到了将代码片段组合的作用
- 局部变量加上关键字global，可令其变为全局变量，即在函数外也可以访问

通过函数实现递归

- 斐波那契数列
- x(1)=1;
- x(2)=1;
- X(n)=X(n-1)+x(n-2), n>2

```php
$n = 100;
$x=array();
$x[1]=1;
$x[2]=1;
for($i=3; $i<=$n; $i++) {
    $x[$i] = $x[$i-1] + $x[$i-2];
    echo "x($i)=".$x[$i].'<br>';
}
```

```php
function my_fabo($num) {
    if ($num == 1) {
        return 1;
    }elseif ($num == 2) {
        return 1;
    }else{
        return my_fabo($num-1) + my_fabo($num-2);
    }
}

$n = 10;
echo "x($n)=".my_fabo($n);
```

作业一

用递归函数实现汉诺塔如何移动

作业二

用函数实现矩阵的加减乘除转置操作